用户故事 | 曾轼麟：世上无难事，只怕有心人

2021-10-07 曾轼麟

《Redis源码剖析与实战》 课程介绍 >

讲述：王惠

时长 11:32 大小 10.57M

你好，我叫曾轼麟，目前在广州一家互联网公司，从事互动社交业务领域中的服务端开发工作。

我从《Redis 核心技术与实战》的时候，就开始跟随蒋老师的脚步，一直在学习 Redis 相关的知识。在这个过程中，我收获了很多新的认知，也对 Redis 的使用及其底层实现有了不一样的理解，所以，很高兴能在这里跟你分享我学习课程的心得与体会、学习思路与方法。

为什么学习 Redis 源码？

其实早在老师开设这门课程之前，我就已经坚持阅读 Redis 源码一段时间了。

一开始我有想去阅读 Redis 源码的念头时，应该是在 2020 年初，那时候 Redis 6.0 版本新增了 IO 多线程的这个特性。而一直以单线程设计为主推的 Redis，突然高调引入了多线程的设计，就让我对其新的设计思路感到非常好奇。

在随后的一段时间里，Redis 社区也发生了重大的变化：安迪斯（Antirez，Redis 之父）不再维护 Redis，而是全权交给了社区进行维护。由此 Redis 开源的方式就由专制模式，转变为了社区自治模式。老实说，我其实挺期待这种变化会给 Redis 的未来带来什么样的影响。也是因为这种种原因，就促使我开展了源码阅读之旅。

到了 2021 年的 1 月份，公司上市后为了进一步抢占市场，领导和我们提出：市场今年需要发力，本年度的 OKR 可能会是 5 倍、10 倍乃至 30 倍的 DAU 增长，而面临这样的增速，我认为对系统性能的优化要求就迫在眉睫。并且年初的时候由于 DAU 的增长，几乎每隔一天晚上，我都会被告警叫起来处理线上问题（我们做的是海外业务），在经历了半年的重构和优化后，目前已经趋于稳定。而在这半年的时间里，**无论是研制新的缓存组件、改造现有缓存方案，还是新业务的发展，几乎都离不开 Redis**，甚至连运营和产品同学都会问我们，这次设计方案咱用 Redis 吗？

所以，除了我个人对于 Redis 的新特性和底层实现原理的关注之外，在工作上的各种变化和挑战，也驱动我想要快速提升阅读 Redis 源码的能力，并能通过读源码的设计思路，帮助指导和解决我在优化高性能接口上遇到的各种难题。

我是怎么学习 Redis 源码的？

那么，我是怎么学习源码的呢？

其实在一开始阅读 Redis 源码的时候，我也走了一些弯路，无论是从 IDE 的选择，还是确定代码阅读的切入点，我都花费了大量的时间，但最后我发现花费了时间却没达到理想的效果。所以这里，我先来分享一些我踩坑后总结的经验 / 建议，希望能通过这些经验 / 建议，帮助你更好地阅读源码。

Linux

首先，就是一定要有一套 Linux 的环境，它能自行编译打包源码，从而可以方便我们做调试，（Windows 的同学可以使用虚拟机），这里可以参考 Redis 在 GitHub 上面的文档。我曾经尝试在 Windows 去编译 Redis，但是最后发现成本比想象中的要高，而且

Windows 对编译环境不太友好，导致编译出来验证后的效果，不一定是和实际服务器的效果一致的。

Clion

为了方便阅读源码，我曾经选择过一些 IDE 工具来作为辅助，比如 VS Code、Visual Studio，甚至是文本工具，但是最后为了保留 Java 开发者的快捷键习惯，我最后是选择使用了 CLion 这款软件。所以，如果你也是 Java 开发者，这里我建议你可以考虑把源码阅读环境和调试环境分离开，这样就可以避免很多不必要的环境冲突。比如在 Windows 下阅读代码，然后提交到虚拟机中的 Linux 进行断点调试。

GDB

关于 Redis 的调试，我非常推荐这款调试工具，它本身在 Linux 上就已经是兼容的了，每次我需要分析某个指针对象或者缓冲区空间内容的时候，都可以使用它，它甚至还能伪造一些阻塞的场景，比如下图是我在 getGenericCommand 方法入口进行断点，并输出 Client 结构体里面 querybuf 的内容（图中是 RESP 协议的字符串）。

```
127.0.0.1:6379> get test
```

```
(gdb) p c->querybuf
$1 = (sds) 0x7fffedffce45 "*2\r\n$3\r\nget\r\n$4\r\ntest\r\n"
(gdb)
```

不要忽略注释

在 Redis 的源码中存在着很多注释，尤其是那些写在某一行代码上面的注释，这些注释很多时候能最直接代表作者的想法和思路，所以我认为这些注释是一定要翻译细读的，可以减少很多我们在阅读源码过程中的疑惑。比如在《23 | 从哨兵 Leader 选举学习 Raft 协议实现（上）》的每课一问中，蒋老师提出的问题就能在如下的注释中得到解答：

复制代码

```
1 void sentinelTimer(void) {
2     sentinelCheckTiltCondition();
3     sentinelHandleDictOfRedisInstances(sentinel.masters);
4     sentinelRunPendingScripts();
```

```
    sentinelCollectTerminatedScripts();
    sentinelKillTimedoutScripts();
    //这里就是 server.hz 需要动态调整作者为此做出的解释
    /* We continuously change the frequency of the Redis "timer interrupt"
     * in order to desynchronize every Sentinel from every other.
     * This non-determinism avoids that Sentinels started at the same time
     * exactly continue to stay synchronized asking to be voted at the
     * same time again and again (resulting in nobody likely winning the
     * election because of split brain voting). */
    server.hz = CONFIG_DEFAULT_HZ + rand() % CONFIG_DEFAULT_HZ;
}
```

结合官网文档和 GitHub 的文档

很多疑问和设计思路其实在这些地方都可以找到相应的解答，比如GitHub上面，我们可以找到 Redis 编译、配置相关的文档。而在官网，我们可以了解到 Redis 的各种设计思想等。在学习课程的过程中，我认为还可以回到社区，去时常关注 Redis 正在发生的变化，这样来结合源码的解读和学习，就能帮助我们获得更好的学习效果。

而更重要的就是，刚刚开始阅读的同学，一定要跟随老师的脚步，因为老师会按照源码结构和知识点的内容划分模块，并通过课程讲述的逐渐推进，帮我们找到每个功能模块合适的切入点，这样能避免走很多弯路。我在自己阅读源码的时候，其实就是像瞎子过河一样，导致走了弯路，直到看完老师的课程才豁然开朗。

基础知识很重要

在学习和阅读 Redis 源码期间，我能很明显地感受到基础知识的重要性。**万丈高楼平地起**，无论是 Redis 中的跳表、Hash 表的实现，还是 Redis 中多线程与多进程的协作，几乎都离不开基础知识。这里，我来推荐两本书和一个课程，这些内容对我理解 Redis 设计原理和阅读 Redis 源码帮助很大。

《深入理解计算机系统》

CSAPP 想必很多人都有所接触，这里我以第三版为例，书中的第八章到第十二章节中，就涵盖了 Redis 实现相关的基础知识内容，比如进程、虚拟内存、系统 I/O、网络、并发编程，等等。比如：在 Redis 源码中使用的 fork() 在书中《进程》章节就有专门说明，还有写时复制形成原因，在《虚拟内存》的章节中也有。

《算法导论》

可能很多人听到这个名字就感到害怕，但这本其实并不是每个章节都是晦涩难懂的，就以第三版为例，书中第三部分“数据结构 - 散列表”，以及第五部分“高级数据结构 -B 树”，在我们的这门 Redis 源码课的学习过程中，也时有提及。

徐文浩老师的课程《深入浅出计算机组成原理》

在这门课程中，老师提到了内存伪共享、CPU 缓存行和虚拟内存等相关知识，也有各种计算机结构相关的知识点。读完也许你就能明白：为什么 Redis 会一直坚持单线程的设计思路？以及为什么以前我们经常会建议，在部署 Redis 的时候进行绑核操作？

当然我这里也不是说，我们需要全部读完，而是可以结合着来学习，当我们遇到某些不太容易掌握的知识点的时候，可以尝试着去这些书籍课程中寻找相关的补充资料，这样更有助于深入理解 Redis 源码的设计初衷。

实践和回馈也很重要

读万卷书不如行万里路，对 Redis 的学习也一样，我们不能只阅读源码，还需要经常去做尝试（当然这里我不建议直接在线上尝试）。**只有不断地试错，才能得到成长。**

我曾经就在线下，使用过不同的内存分配机制来编译 Redis，最后通过压测工具去对比效果。或者是在 Redis 中，通过断点去观察 querybuf 中 RESP 协议的解析过程。在每次的尝试过程中，我都能有所收获，都能感受到动手实操所带来的愉悦感，并且愈发有深入学习下去的动力。

所以我建议你可以结合一些场景去进行实践，可以是对源码断点进行调试，也可以是针对源码的改造，哪怕只是一个日志输出，再或者是结合实际的业务场景做出一些改进，做出点成果，然后用这些成果来回馈自己。

而回到 Redis 源码的课程中，对于**每课一问**，我也是非常期待的，因为我总能看到一些同学有不一样的想法和观点，或者提出一些新颖的提问。在这个交流的过程中，其实也是一种学习的回馈。老师的提问，有助于引导我们去思考 Redis 这样设计的价值和意义。

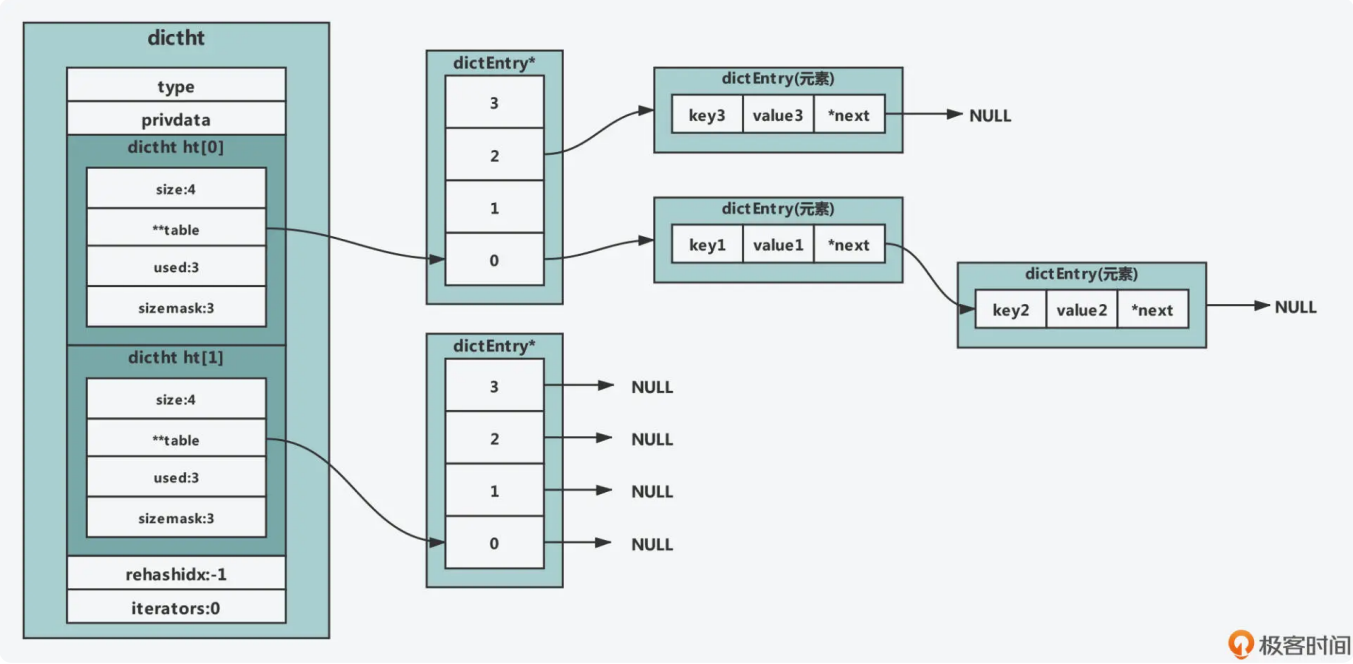
死磕精神和知识沉淀

阅读源码的过程一定会伴随着枯燥，尤其是当遇到一个知识盲区，或者是自身无法理解的事物的时候，这种知识上的挑战所带来的焦虑的情绪，就会导致我们失去了继续看下去的动力。而每到这种时候，我就会让自己休息一下，换个心情，说不定第二天就想通了。

我还记得当时在阅读 Redis 生命周期相关代码的时候，我就整整花了一周的时间，去画出整个生命周期的时序图，最终才把整个流程和思路走通。后来我渐渐发现，每当我想起一个 Redis 的功能特性，几乎都能有办法迅速定位到源码的位置，而这就是我通过转换心情和坚持做事的死磕精神，给我带来的学习收益。

同时，在阅读 Redis 源码的过程中，我也会输出自己的知识文档 / 博客，每次阅读完某个功能的代码后，都会记录下当时的理解过程和思路。然后，每当我重新阅读这部分代码的时候，就会回来作对比，结果往往就会发现，随着我的知识面的拓展和对代码的深入理解，很多细节其实和当时的理解是有偏差的，那么这个时候我就会去修正文档，而这样对我自己来说，其实也是一种知识的沉淀和积累。

比如，下面展示的就是我之前所做的辅助理解图例。



字典数据结构图

写在最后

最后，很高兴能和大家一起学习 Redis 源码，并且能有机会分享自己的学习心得。

我觉得，订阅这门课程的同学，本身就是对技术有更高追求的，对技术都是有前瞻视野的。虽然 Redis 源码的学习是一个持续且漫长的过程，需要付出时间去坚持，我们可能一开始会备受打击，但是一旦学有所成，它带给我们的收获，一定是非常有价值的。

世上无难事，只怕有心人，祝大家在学习完课程后，自身的能力都能更上一层楼。也期待各位的留言，我们一起交流和讨论，一起学习，共同进步。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 3 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 加餐4 | RDB和AOF文件损坏了咋办？

下一篇 26 | 从Ping-Pong消息学习Gossip协议的实现

精选留言 (1)

写留言

zhangm365

2021-10-08

谢谢分享，真的强，我直接跟着蒋老师干源码，这过程中看大佬的分享收获很多。如果蒋老师能多在评论区互动解答就好了

展开

1 1